A few applications of the SVD

Many methods require to approximate the original data (matrix) by a low rank matrix before attempting to solve the original problem.

- Regularization methods require the solution of a least-squares linear system $Ax = b$ approximately in the dominant singular space of $A$.

- The Latent Semantic Indexing (LSI) method in information retrieval, performs the “query” in the dominant singular space of $A$.

- Methods utilizing Principal Component Analysis, e.g. Face Recognition.

Commonality: Approximate $A$ (or $A^\top$) by a lower rank approximation $A_k$ (using dominant singular space) before solving original problem.

- This approximation captures the main features of the data while getting rid of noise and redundancy.

Note: Common misconception: ‘we need to reduce dimension in order to reduce computational cost’. In reality: using less information often yields better results. This is the problem of overfitting.

Good illustration: Information Retrieval (IR).

Information Retrieval: Vector Space Model

- Given: a collection of documents (columns of a matrix $A$) and a query vector $q$.

- Collection represented by an $m \times n$ term by document matrix

- Queries (‘pseudo-documents’) $q$ are represented similarly to a column

- Problem: find a column of $A$ that best matches $q$.

- Similarity metric: angle between the column and $q$ - Use cosines:

- To rank all documents we need to compute

- $s = A^\top q$

- $s$ = similarity vector.

- Literal matching – not very effective.
Use of the SVD

Many problems with literal matching: polysemy, synonymy, ...

Need to extract intrinsic information – or underlying “semantic” information –

Solution (LSI): replace matrix \( A \) by a low rank approximation using the Singular Value Decomposition (SVD)

\[
A = U \Sigma V^T \rightarrow A_k = U_k \Sigma_k V_k^T
\]

\( U_k \): term space, \( V_k \): document space.

Refer to this as Truncated SVD (TSVD) approach

New similarity vector:

\[
s_k = A_k^T q = V_k \Sigma_k U_k^T q
\]

Issues:

- Problem 1: How to select \( k \)?
- Problem 2: computational cost (memory + computation)
- Problem 3: updates [e.g. google data changes all the time]
- Not practical for very large sets

LSI: an example

%% D1 : INFANT & TODLER first aid
%% D2 : BABIES & CHILDREN’s room for your HOME
%% D3 : CHILD SAFETY at HOME
%% D4 : Your BABY’s HEALTH and SAFETY
%% D5 : From INFANT to TODDLER
%% D6 : BABY PROOFING basics
%% D7 : Your GUIDE to easy rust PROOFING
%% D8 : Beanie BABIES collector’s GUIDE
%% D9 : SAFETY GUIDE for CHILD PROOFING your HOME

%% 6:INFANT 7:PROOFING 8:SAFETY 9:TODDLER
%% Source: Berry and Browne, SIAM., '99

Number of documents: 8
Number of terms: 9

Raw matrix (before scaling).

\[
A = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Get the answer to the query Child Safety, so

\[
q = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]
\]

using cosines and then using LSI with \( k = 3 \).
**Dimension reduction**

Dimensionality Reduction (DR) techniques pervasive to many applications

- Often main goal of dimension reduction is not to reduce computational cost. Instead:
  - Dimension reduction used to reduce noise and redundancy in data
  - Dimension reduction used to discover patterns (e.g., supervised learning)

- Techniques depend on desirable features or application: Preserve angles? Preserve distances? Maximize variance? ..

**The problem**

- Given $d \ll m$ find a mapping $\Phi : x \in \mathbb{R}^m \rightarrow y \in \mathbb{R}^d$
- Mapping may be explicit (e.g., $y = V^T x$)
- Or implicit (nonlinear)

**Practically**: Find a low-dimensional representation $Y \in \mathbb{R}^{d \times n}$ of $X \in \mathbb{R}^{m \times n}$.

- Two classes of methods: (1) projection techniques and (2) nonlinear implicit methods.

**Example: Digit images (a sample of 30)**

![Digit images](image)

**A few 2-D 'reductions':**

- PCA - digits: 0 --- 4
- LLE - digits: 0 --- 4
- K-PCA - digits: 0 --- 4
- ONPP - digits: 0 --- 4
**Projection-based Dimensionality Reduction**

*Given:* a data set $X = [x_1, x_2, \ldots, x_n]$, and $d$ the dimension of the desired reduced space $Y$.

*Want:* a linear transformation from $X$ to $Y$

\[
X \in \mathbb{R}^{m \times n}, \quad V \in \mathbb{R}^{m \times d}, \quad Y = V^T X
\]

$m$-dimens. objects $(x_i)$ ‘flattened’ to $d$-dimens. space $(y_i)$

*Problem:* Find the best such mapping (optimization) given that the $y_i$’s must satisfy certain constraints

**Principal Component Analysis (PCA)**

- PCA: find $V$ (orthogonal) so that projected data $Y = V^T X$ has maximum variance
- Maximize over all orthogonal $m \times d$ matrices $V$:
  \[
  \sum_i \|y_i - \frac{1}{n} \sum_j y_j\|_2^2 = \cdots = \text{Tr} [V^T \bar{X} \bar{X}^T V]
  \]
  
  Where: $\bar{X} = [\bar{x}_1, \ldots, \bar{x}_n]$ with $\bar{x}_i = x_i - \mu, \mu =$ mean.

  **Solution:**
  
  $V = \{\text{dominant eigenvectors}\}$ of the covariance matrix
  
  i.e., Optimal $V =$ Set of left singular vectors of $\bar{X}$ associated with $d$ largest singular values.

**Matrix Completion Problem**

Consider a table of movie ratings. You want to predict missing ratings by assuming commonality (low rank matrix).

<table>
<thead>
<tr>
<th>given data</th>
<th>predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>movie</td>
<td>Paul</td>
</tr>
<tr>
<td>Title-1</td>
<td>-1</td>
</tr>
<tr>
<td>Title-2</td>
<td>4</td>
</tr>
<tr>
<td>Title-3</td>
<td>-3</td>
</tr>
<tr>
<td>Title-4</td>
<td>$x$</td>
</tr>
<tr>
<td>Title-5</td>
<td>3</td>
</tr>
<tr>
<td>Title-6</td>
<td>-2</td>
</tr>
</tbody>
</table>

\[
\text{Minimize } \|(X - A)_{\text{mask}}\|_F^2 + 4\|X\|_*
\]

"minimize sum-of-squares of deviations from known ratings plus sum of singular values of solution (to reduce the rank)."